

# Elements

Each *element* used in the *workflow* must be described inside the body. An element description consists of the element name and a set of parameters enclosed in curly braces. A parameter and the value are separated by ':', different parameters are separated by ';':

```
element_name {  
  
    parameter1:value1;  
    parameter2:value2;  
    ...  
}
```

See, for example, a description of the [Read alignment](#) element:

```
read-msa {  
    type:read-msa;  
    name:"Read alignment";  
    url-in:/home/user/pkinase.sto;  
}
```

Note, that the values of the parameters for an element can also be presented in the [iterations](#) block. For all elements the following parameters are defined:

- **type** - specifies the type of the element.
- **name** - specifies the name of the element. It corresponds to the element's name in the GUI
- **.validator** - validates the element by the input validator type's parameters:
  - **type** - specifies the type of the validator.

For example this validator validate that the read sequence element has two or three datasets:

```
read-sequence {  
    type:read-sequence;  
    name:"Read Sequence";  
    .validator {  
        type:datasets-count;  
        min:2;  
        max:3;  
    }  
}
```

For *custom elements* there is special parameter:

- **script** - sets the script text of the element, for example:

```
dump-info {  
    type:"Script-Dump sequence info"  
    name:"Dump sequence info"  
    script {  
        out_text=getName(in_sequence) + ": " + size(in_sequence);  
    }  
}
```

The list of parameters available depend on an element. Refer to the [Workflow Elements](#) chapter to find out the parameters for a particular element. To [set a script value for a parameter](#) use the following form:

```
parameter_name {  
    a script value  
};
```